BR115188 (2)

AD-A229 274

RSRE
MEMORANDUM No. 4405

ROYAL SIGNALS & RADAR ESTABLISHMENT

NETWORK STABILITY UNDER VIRAL ATTACK – A GAME ENTITLED "GOD AND THE DEVIL"

Author: Dr S C Giess

PROCUREMENT EXECUTIVE,
MINISTRY OF DEFENCE,
RSRE MALVERN,
WORCS.

90 11 27 042

RSRE MEMORANDUM No. 4405

Royal Signals and Radar Establishment

Memorandum 4405

Title: Network stability under viral attack
- A game entitled " God and the Devil"

Author: S. C. Giess

Date: July 1990

Summary

The problem of viral attacks on communications networks is explored with an emphasis on the time taken for network recovery. The problem is studied by means of a game which allows for different initial states of the network followed by state evolution according to a fixed rule-set. It is found that only a small percentage of nodes in the network need be equipped with countermeasure software for a near asymptotic recovery time to be achieved. However, for all realistic countermeasure percentages, the time for complete recovery can be long compared to the time for majority recovery - a situation similar to human illness/recouperation time ratios.

Contents

THIS PAGE IS LEFT BLANK INTENTIONALLY

1.0 Introduction

Recently the topic of "Viruses" infecting computer systems has received attention in the press. Various "Antidotes" and "Filters" have been developed and used. However computer based packet switching communication networks have also been attacked. Such attacks cripple the operation of these networks and usually result in the complete failure of a part or all of the system

The work described in this memorandum is part of a study into the stability of communications networks . Here we explore one facet of the topic; namely what happens if a malevolent "virus" is injected into an otherwise stable network, where the network has countermeasure "antidotes" located at a few points within it.

Now the precise details of the virus and possible antidotes will depend upon the particular communication protocols and the operating systems of the computer nodes. Nevertheless the propagation of such entities through a network may be amenable to study without knowlege of their detailed structure. The only structural assumption is that the virus must affect the network and the antidote must be able to detect and affect the virus . This high level approach is possible because of the limited nature of their goals (for the virus - the destruction of the network, for the antidote - the destruction of the virus and the resuscitation of the network) and the strategies used to attain them (interaction with the nearest connected neighbour).

The investigation takes the form of a game between two principal players. The purpose of the game is to explore the temporal evolution of the state of a network of connected nodes. The goal of one player is to stop the network functioning, the goal of the other is to keep the network fully operational. The game players are only allowed to specify the initial state of the network. After that the game proceeds by the application of a predetermined set of rules which change the state of the network. The game is deemed to have finished when there have been no changes in the network after a complete round.

As will be seen later certain static data can be extracted from playing such a game. However the medium of this memorandum cannot convey satisfactorily the dynamic aspects of some of the evolving node state configurations. This can only really be appreciated by watching the game on a computer display.

2.0 Other network simulations

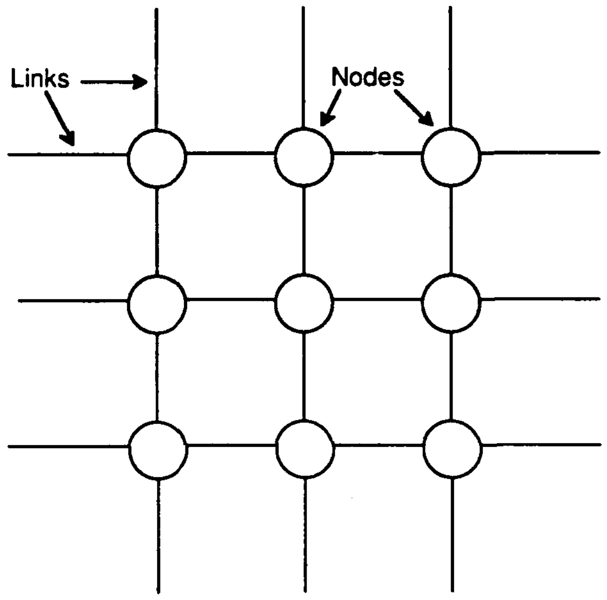
Networks of entities which can interact with their neighbours according to a fixed set of rules are often called cellular automata. Interest has centred on the spatial evolution of inital configurations of the entities. and many networks working to different rule sets have been studied [1,2]. One of the earliest and most famous of these, Conway's game of "Life" [3], partially inspired the exploration described in this memorandum; with the expectation that the results would not be too dissimilar from those found by Conway's rule set. Briefly Conway discovered that whilst some initial configurations of cells would only "live" for a short while, others developed structures that were long lived. Indeed some could propagate around the network maintaining their relative configuration. These could exist for a long time before dying out. Other configurations produced structures which had a cyclic nature resulting in them persisting for ever. Other simulation work [2] has shown that networks of interacting biological entities can attain states which persist for long timescales, creating Evolutionary Stable Systems (ESSs).

Of course such temporal and configurational properties depend upon the set of rules which govern the interaction of the entities, so no claim is made that there will be exact analogies for a rule set based upon communication principles. Indeed communication networks often comprise entities which interact with each other at several different levels of abstraction, with each level having a different rule set. Nevertheless there is enough of a similarity with simple cellular automata to ask whether an instability in a communications network, once caused by whatever means, will actually decay and how long it will take for this to happen.

3.0 The model network

The network (Fig 1) comprises nodes, which are points where information flows come together or diverge, and links, which are the paths along which the information can travel.

Fig 1



3.1 The structure of the network.

For ease of study a grid connection pattern has been chosen for the network links where each node is in direct connection with four neighbours only. Each link is bidirectional. The sources and sinks of information which would usually be connected to the nodes have been neglected.

The assumption is made that it is possible to send instructions from one node to another which can turn on or off the recipient node's capability to handle data. We make no assumption as to whether a normal user of the network can send such instructions via the usual data terminals, or whether there are separate control channels which are only accessable to priviliged users, perhaps on separate hardware circuits.

3.2 The allowable states for a node

We assume that a node can be in one of four states. These are :

1) Dead. The node does nothing!

2) Ordinary. The node is doing its desired job of communicating with its neighbours, so sending data around the network.

3) Daemonic. The ordinary operation of the node has been subverted. Now the node has only one goal which is to subvert its ordinary neigbours to do the same actions as itself and then to disconnect itself from its neighbours and die, so fragmenting the network.

4) Angelic. The ordinary operation of the node continues but the node checks on the condition of its neighbours. If they are not operating normally then the node is capable of resetting the abnormal nodes and moving its correcting capability to those nodes, leaving an ordinary node behind.

The reader will note that theological names have been chosen. This is for the following reason: The names "virus" and "antidotes" imply a medical perspective on the matter, really with connotations of a disinterested problem. However these viruses were not created by chance mutation, rather they were created by people with deliberate intent to do evil. Likewise the antidotes are deliberately created to save the network from the malevolent viruses, indeed one can view an antidote as a benign virus! Given these points, it seems that a more theological naming is appropriate. On one side we have the creator of problems, the Devil, with his assistant daemons. On the other side we have the curer of problems, God, with his assistant angels.

3.3 The allowable states of the links

The properties of the links are assumed to be determined by the nodes to which they are connected. That is, a link is operational only if both the nodes are operational. Conversely a 'ink can only be broken by one or either of the nodes failing - we do not allow links to be affected directly, for example by being cut.

4.0 The players

There are two principal players, God and the Devil. The only parts they play in the proceedings are to initialise the network with their assistants, after that they stand back and watch the ensuing changes which arise solely from the actions of their assistants.

4.1 The rules governing the players

4.1.1 The Devil's rule.

Initialise some network nodes to the daemonic state.

4.1.2 The rules for each daemon:

First examine the state of each nearest neighbour node.

Next for each neighbour:

1) If it is daemonic, do nothing.

2) If it is " ordinary", spawn a daemon to it.

3) If it is dead, do nothing.

4) If the node is angelic, do nothing.

Finally kill the node upon which the daemon is sitting.

4.1.3 God's rule.

Initialise some network nodes to the angelic state.

4.1.4 The rules for each angel:

First examine the state of each nearest neighbour node.

Next for each neighbour:

1) If it is either "ordinary" or angelic, do nothing.

2) If it is daemonic, kill the daemon and spawn an angel to it.

3) If it is dead then resuscitate it spawning an angel to it.

Finally If either of the tests in rules 2) or 3) have been successful then vacate the present node, leaving it as an ordinary node.

4.2 Justification for the assigned functionalities

We assume that the attack upon this network is mounted by an individual or organisation - the Devil. We assume that the possibility of such an attack has been forseen and that an individual or organisation has the job of combating it - God. The daemons are merely the disruptive instructions which are infiltrated at the initial weak points. The angels are the corrective software instructions. We assume that such corrective actions reduce the ordinary throughput of a node and so it is not desirable to have too many in the network under normal conditions. We note that if there were no overhead penalty then the simplest solution would for every node to be angelic.

We assume that the corrective instructions cannot be subverted: hence the angels cannot be killed. However we assume that the subversive instructions can be eliminated, hence the angels ability to kill a daemon. Of course it is moot whether, in reality, a daemon could not kill an angel!

5.0 **Simulation aspects**

5.1 Rules are applied in sequence, first the Devil's turn then God's turn. After that the daemons have their turn followed by the angels turn. To simplify matters we assume that the network works in a synchronised manner, as if there were a global clock. Hence we can conceive of the network as being in a state M at time N which, with one tick of the global clock, is transformed into state M+1 at time

N+1. Here the states ignore the data content in the network. Of course the rules governing these state changes determine the temporal evolution of the network. We note that this is a time sliced approach to the problem of modelling what in reality would be an asynchronous parallel problem.

5.2 The distribution of initial daemonic states and angelic states is random. This is justified as follows:

We assume that given the large size of networks some nodes may be more easily infiltrated than others and so the initial attack will come from these nodes. Of course if the defender knew which nodes were weak then they could better protected, but we assume that in an imperfect world such knowledge does not in practice exist. Hence as far as the defender is concerned the initial attack will come from unexpected nodes, so the best he can do is to distribute his correcting instructions at random. The initial percentages of the distributions are specified by the players. Note: the actual node state allocations for the players were completely independent of each other and could overlap. In the event of an allocation overlap on a node, (ie it was to be both an angel and a daemon) the angelic state was given precedence.

5.3 We note that the particular choice of rules given above, coupled with the timesliced nature of the application of the rules results in a node distribution per turn which is independent of the order in which the nodes are processed. However it is possible to have a different set of rules, whilst still retaining the timesliced feature, that results in an order sensitive node distribution.

6.0 Implementation details

A demonstration programme implementing the rules was run on an Atari 1040 computer, which took advantage of the Atari's colour graphics capability. The four colour, medium resolution graphics mode was used which has a pixel resolution of 607 wide by 166 high. This was sufficient for a displaying a square grid network of size 14 by 14.

Interpreted Basic was used for ease of programming the graphics routines. The use of an interpreted language was no limit on the performance of the programme as the rate of network update was appropriate for direct viewing by people given the sizes of networks that could usefully be displayed.

The implementation of the random seeding was done taking advantage of the flexibility of the Basic (pseudo) random call to be given a user selected seed value. This enabled repeatable runs to be mounted if desired.

7.0 Experiments

Three sets of runs were mounted on a 10 by 10 network.

The first set was composed of a series of runs. In each series the percentage of angels was held constant, whilst the percentage of daemons increased in steps of 5 %. The percentage of angels was increased in steps of 5% between the series. The random number generator used the repeatable option, this meant that for each run of a given series the initial angel pattern was identical. The number of iterations that occured before the network stabilised was recorded along with the cumulative number of attempted node state changes. This last figure gives an indication of the link traffic that such an intruder removal will entail.

The second set comprised a smaller series of runs. This time the initial percentages of daemons and angels were kept constant but the inital seed value to the pseudo random number generator was varied. The aim was to investigate the likely dependence upon the precise initial distribution of entities.

In the third set of runs the number of attempted node changes per iteration were recorded for some selected initial percentages of daemons and angels. The random number generator seed dependence was also investigated.

7.1 Results and Discussion

Owing to the large amount of data produced by the simulation it is not possible to reproduce all here. Instead aspects are highlighted and are given in tables 1 to 19.

The first aspect is a study of both the time, as measured by the number of iterations, and the link traffic, taken to restore a network to full operation as a function of the initial percentages of the entities.

The figures in tables 1 to 6 were calculated with the (pseudo) random number generator's seed value initialised to 1. The percentages of angels and daemons columns are for the inital state. The number of iterations column includes the last iteration for which there is no change in the state of the network. The cumulative change total column is the sum of the number of node changes that each node would make per iteration. Included are the multiple attempted changes that arise when the state of a node can be changed by more than one neighbour. This method of accounting describes more accurately the traffic that would flow when several neighbours try to affect a node at the same time without themselves knowing that their efforts are being duplicated.

We can see that it takes longer for a network to be restored when the percentage of angels is small. However for greater than 10% angels the time reduction is marginal.
We note the tendency for a higher total of link traffic for the runs with the small percentages of angels. However, for a given percentage of angels, the totals are not strongly dependent upon the percentage of daemons

The second aspect is the dependence of the results above on the particular initial seed value.
Table 7 shows what happens when the nominal percentages of angels and daemons are kept constant at 25% and 70% repectively (table 5's values), but a different seed value used for the random number generator. We can see that there is nearly a two to one variation in the time taken to restore the network. This shows a considerable dependence upon the precise spatial distribution of the entities. The attempted link traffic shows less variation. The figures given in tables 8 to 19 show similar dependencies for different angel - daemon percentages.

The third aspect is a study of the recovery rates again as a function of nominal percentage values. As mentioned above seed values dependencies are also examined.
Tables 8 to 11 show the number of attempted node changes per iteration for 25% angel, 70% daemon values. The same initial four seed values used in preparing in table 7 are used. Here. we note that

there is an initial flurry of link traffic decaying quickly to a long weak tail.

Tables 12 to 15 show the random number generator seed dependences and the attempted node changes per iteration for 10% angel, 5% daemon values. Here the traffic starts off low, rises to a long plateau before decaying.

Tables 16 to 19 show the random number generator seed dependences and the attempted node changes per iteration for 10% angel, 70% daemon values. Here the link traffic is initally high before decaying comparatively slowly and linearly to zero.

7.2 General Observations arising from the simulation

7.2.1 Some initial configurations resulted in wavefronts of angels and daemons traversing the network. There were some multiple wavefronts seen as well. The non-wrapround nature of the network boundary caused this phaenomenon to die when the waves encounted the boundary. A wrapround boundary condition would have extended the time for these structures to persist. Such a wavefront phaenomenon implies that the complete recovery time for such unfortunate network configurations will be markedly longer than *ones estimated* from "mean" initial percentages.

7.2.2 During the attack of the daemons the living part of the network could be reduced to a very small size compared to its inital extent. This implies that significant fractions of the users could be entirely without communications, as against just suffering a reduction in network performance.

7.2.3 The time for the system to completely recover (in the sense that all nodes are operational again) after the last daemon died could be of the order of the cumulative time spent chasing and eliminating the daemons. This is analogous to the usual medical illness - recovery/recouperation phaenomenon.

7.2.4 The process of recovery leaves many more angel nodes distributed around the network than were originally seeded. The consequence would be that the network's throughput would be reduced because of the computational overheads associated with the "angel" software.

7.2.5 One might expect the daemons to be at their most devastating per daemon when their initial population density meant that they were just isolated from each other. This way they would infect the greatest number of neighbours per daemon. However a study of the results does not support this hypothesis.

8.0 Conclusions

We have found that our communication network example has indeed stability properties which are similar to those attributed to Conway's Game of Life. We have demonstrated that, for our rule set and network connectivity, only a small percentage (approx 10 %, distributed around the network at random) of nodes need be equipped with countermeasure capability to give as fast a network recovery as would be achieved by a much greater percentage (30%). We also have shown that the time taken to completely clear such a network of an invading force is dominated by the effort to eradicate the last few elements of that force.

9.0 References

[1] A.K. Dewdney, Computer Recreations, Scientific American, Jan 1990, p136-139 (and references therein)

[2] R. Dawkins, ' The Selfish Gene ', Oxford University Press, 1989

[3] M. Gardner, Mathematical Games, Scientific American, Oct 1970, p120-123

10.0 Acknowledgements

I am pleased to thank B. Bramson for his helpful discussions in the early stages of this work and T.A.D. White, C. I. Walmsley and T.B. Dean for comments on the draft document.

Table 1

% of Angels	% of Daemons	Number of iterations	Cumulative change total
5	5	19	546
5	10	19	547
5	15	19	534
5	20	19	534
5	25	18	1079
5	30	21	1734
5	35	24	1846
5	40	24	2015
5	45	18	1072
5	50	16	525
5	55	16	520
5	60	16	513
5	65	16	511
5	70	16	509

Table 2

% of Angels	% of Daemons	Number of iterations	Cumulative change total
10	5	10	527
10	10	12	796
10	15	12	824
10	20	13	760
10	25	9	663
10	30	15	919
10	35	13	1075
10	40	12	661
10	45	12	663
10	50	8	517
10	55	7	463
10	60	7	447
10	65	7	451
10	70	7	449

Table 3

% of Angels	% of Daemons	Number of iterations	Cumulative change total
15	5	8	305
15	10	9	466
15	15	11	583
15	20	9	437
15	25	9	517
15	30	7	494
15	35	10	542
15	40	7	480
15	45	7	482
15	50	7	463
15	55	7	468
15	60	6	419
15	65	6	423
15	70	6	421

Table 4

% of Angels	% of Daemons	Number of iterations	Cumulative change total
20	5	8	206
20	10	10	379
20	15	8	385
20	20	7	340
20	25	7	387
20	30	9	457
20	35	11	445
20	40	6	403
20	45	6	407
20	50	7	399
20	55	7	401
20	60	5	342
20	65	5	361
20	70	5	361

Table 5

% of Angels	% of Daemons	Number of iterations	Cumulative change total
25	5	5	138
25	10	5	171
25	15	4	166
25	20	4	193
25	25	6	299
25	30	6	295
25	35	6	301
25	40	6	313
25	45	6	319
25	50	4	287
25	55	4	287
25	60	4	289
25	65	4	309
25	70	6	337

Table 6

% of Angels	% of Daemons	Number of iterations	Cumulative change total
30	5	7	143
30	10	5	165
30	15	5	165
30	20	5	154
30	25	6	320
30	30	6	276
30	35	6	283
30	40	5	273
30	45	5	272
30	50	5	264
30	55	5	264
30	60	6	282
30	65	6	304
30	70	6	315

Table 7

Random seed value	% of Angels	% of Daemons	Number of iterations	Cumulative change total
1	25	70	6	337
2	25	70	4	347
3	25	70	7	427
4	25	70	6	314
5	25	70	6	392
6	25	70	5	386
7	25	70	5	348
8	25	70	4	410
9	25	70	4	359
10	25	70	7	431

Table 8

Random seed value	% of Angels	% of Daemons	Iteration number	Change per iteration
1	25	70	1	210
			2	94
			3	9
			4	13
			5	11
			6	0

Table 9

Random seed value	% of Angels	% of Daemons	Iteration number	Change per iteration
2	25	70	1	230
			2	102
			3	15
			4	0

Table 10

Random seed value	% of Angels	% of Daemons	Iteration number	Change per iteration
3	25	70	1	216
			2	111
			3	35
			4	34
			5	17
			6	14
			7	0

Table 11

Random seed value	% of Angels	% of Daemons	Iteration number	Change per iteration
4	25	70	1	232
			2	58
			3	12
			4	8
			5	4
			6	0

Table 12

Random seed value	% of Angels	% of Daemons	Iteration number	Change per iteration
1	10	5	1	26
			2	62
			3	87
			4	87
			5	90
			6	58
			7	50
			8	53
			9	14
			10	0

Cumulative change total = 527

Table 13

Random seed value	% of Angels	% of Daemons	Iteration number	Change per iteration
2	10	5	1	30
			2	30
			3	29
			4	51
			5	20
			6	4
			7	0

Cumulative change total = 164

Table 14

Random seed value	% of Angels	% of Daemons	Iteration number	Change per iteration
3	10	5	1	24
			2	65
			3	76
			4	80
			5	58
			6	28
			7	6
			8	0

Cumulative change total = 337

Table 15

Random seed value	% of Angels	% of Daemons	Iteration number	Change per iteration
4	10	5	1	28
			2	56
			3	75
			4	87
			5	55
			6	20
			7	60
			8	0

Cumulative change total = 381

Table 16

Random seed value	% of Angels	% of Daemons	Iteration number	Change per iteration
1	10	70	1	175
			2	124
			3	63
			4	61
			5	22
			6	4
			7	0

Cumulative change total = 449

Table 17

Random seed value	% of Angels	% of Daemons	Iteration number	Change per iteration
2	10	70	1	199
			2	142
			3	70
			4	24
			5	38
			6	26
			7	0

Cumulative change total = 499

Table 18

Random seed value	% of Angels	% of Daemons	Iteration number	Change per iteration
3	10	70	1	185
			2	141
			3	69
			4	50
			5	18
			6	0

Cumulative change total = 463

Table 19

Random seed value	% of Angels	% of Daemons	Iteration number	Change per iteration
4	10	70	1	185
			2	133
			3	77
			4	52
			5	32
			6	0

Cumulative change total = 479

THIS PAGE IS LEFT BLANK INTENTIONALLY

REPORT DOCUMENTATION PAGE

DRIC Reference Number (if known) ....

Overall security classification of sheet ....Unclassified. ....

(As far as possible this sheet should contain only unclassified information. If it is necessary to enter classified information, the field concerned must be marked to indicate the classification eg (R), (C) or (S).

Originators Reference/Report No. MEMO 4405	Month JULY	Year 1990
Originators Name and Location RSRE, St Andrews Road Malvern, Worcs WR14 3PS		
Monitoring Agency Name and Location		
Title NETWORK STABILITY UNDER VIRAL ATTACK - A GAME ENTITLED "GOD AND THE DEVIL"		
Report Security Classification UNCLASSIFIED	Title Classification (U, R, C or S) U	
Foreign Language Title (in the case of translations)		
Conference Details		
Agency Reference	Contract Number and Period	
Project Number	Other References	
Authors GIESS, S C	Pagination and Ref 21	

Abstract

The problem of viral attacks on communications networks is explored with an emphasis on the time taken for network recovery. The problem is studied by means of a game which allows for different initial states of the network followed by state evolution according to a fixed rule-set. It is found that only a small percentage of nodes in the network need be equipped with countermeasure software for a near asymptotic recovery time to be achieved. However, for all realistic countermeasure percentages, the time for complete recovery can be long compared to the time for majority recovery - a situation similar to human illness/recuperation time ratios.

Abstract Classification (U,R,C or S)
U

Descriptors

Distribution Statement (Enter any limitations on the distribution of the document)

UNLIMITED

S80/48

THIS PAGE IS LEFT BLANK INTENTIONALLY